

What is claimed is:

1. An automatic compiler test program generation method, comprising the steps of:

5 selecting a plurality of sub-procedural descriptions by a random number-used method from a set of sub-procedural descriptions each of which may form an element of a compiler test program; and

 combining the selected sub-procedural descriptions to
10 generate a compiler test program.

 2. An automatic compiler test program generation method according to claim 1, wherein each sub-procedural description is assigned a value which indicates the probability of the sub-procedural description being selected and the value is
15 reflected in the random number-used selection of sub-procedural descriptions.

 3. An automatic compiler test program generation method according to claim 1, wherein variables included in the selected plural sub-procedural descriptions are made shared
20 by the selected sub-procedural descriptions when the selected sub-procedural descriptions are combined.

 4. An automatic compiler test program generation method according to claim 1, wherein each selectable sub-procedural description internally has a procedure to display a result

of having a computer execute the sub-procedural description on a display unit of the computer and/or a procedure to store the result in a storage unit of the computer.

5. An automatic compiler test program generation method,
5 comprising the steps of:

selecting a plurality of program cells by a random number-used method from a set of program cells each of which may form an element of a test program for a compiler after compiled by the compiler; and

10 combining the selected program cells to generate a test program;

wherein each program cell has a description which specifying whether the program cell is to be compiled alone or in combination with the other selected program cells.

15 6. An automatic compiler test program generation method according to claim 5, wherein the set of program cells includes at least a basic cell where procedures are listed in the order of execution, a control cell (repeat cell) described to repeat a procedure, a control cell (select cell) described to select
20 a procedure and a function cell described to call a function.

7. An automatic compiler test program generation method according to claim 6, wherein the selected program cells are registered with a table in the order of selection and the registered cells are given the following combining process

in the order of registration:

(1) if the cell preceding the current program cell is said basic cell or said function cell, the current cell is concatenated to the preceding cell; and

5 (2) if the cell preceding the current program cell is said control cell, the current program cell is nested into the preceding cell.

8. An automatic compiler test program generation method according to claim 7, wherein the variables included in the
10 selected plural program cells are made sharable among the selected program cells when the selected program cells are combined by the combining process.

9. A compiler test system comprising:

test program generation means for generating a test
15 program;

testing means for taking in the generated test program, compiling the test program by using a test target compiler and outputting an execution result;

expected value generating means for taking in the
20 generated test program, compiling the test program by using a proven compiler and outputting an expected value; and

result comparison means for taking in the execution result and the expected value for comparison with each other and storing the comparison result in a storage unit as a test

result.

10. A compiler test system according to claim 9, wherein the test program generation means selects a plurality of sub-procedural descriptions by a random number-used method
5 from a set of sub-procedural descriptions each of which may form an element of a test program for a compiler, and combines the selected sub-procedural descriptions to generate a test program.

11. A compiler test method comprising the steps of:
10 generating a test program;

taking in the generated test program, compiling the test program by using a test target compiler and outputting an execution result;

taking in the generated test program, compiling the
15 test program by using a proven compiler and outputting an expected value;

taking in the execution result and the expected value for comparison with each other; and

storing the comparison result in a storage unit as a
20 test result.

12. A compiler test method according to Claim 11, wherein the step of generating a test program comprises the sub-steps of:

selecting a plurality of sub-procedural descriptions

by a random number-used method from a set of sub-procedural descriptions each of which may form an element of a test program for a compiler; and

combining the selected sub-procedural description to
5 generate a test program.

13. A program product for allowing a computer to execute a functional verification process for a compiler, said functional verification process comprising the steps of:

selecting a plurality of sub-procedural descriptions
10 by a random number-used method from a set of sub-procedural descriptions each of which may form an element of a test program for a compiler;

combining the selected sub-procedural descriptions to generate a test program;

15 taking in the generated test program, compiling the test program by using a test target compiler and outputting an execution result;

taking in the generated test program, compiling the test program by using a proven compiler and outputting an
20 expected value;

taking in the execution result and the expected value for comparison with each other; and

storing the comparison result in a storage unit as a test result.